

AMENDMENTS

IN THE CLAIMS:

Please amend claims 1, 2, 4, 7, 9, 10, 12, 14 and 16-21 as follows:

1. (Currently Amended) A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method steps comprising:

~~providing~~ implementing an interface for communication with a demultiplexer object, the interface taking ~~which takes~~ multiplexed multimedia data as input and outputting demultiplexed elementary media streams, the interface comprising including:

an Initialize method ~~to configure~~ configuring the demultiplexer object;

a SetPresentationDescriptor method ~~to dynamically setting~~ an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists ~~merely only~~ if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method ~~to provide~~ providing a new input muxed stream to the demultiplexer object;

a ProcessOutput method to ~~retrieve~~ retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor; and

a Flush method to ~~flush~~ flushing currently queued input and output samples.

2. (Currently Amended) The computer-readable storage medium of claim 1 wherein the interface ~~further~~ comprises a GetPresentationDescriptor method to retrieve a clone of the currently active presentation descriptor on the demultiplexer object.

3. (Previously Presented) The computer-readable storage medium of claim 2 wherein the GetPresentationDescriptor method includes a presentation descriptor.

4. (Currently Amended) The computer-readable storage medium of claim 1 wherein the interface ~~further~~ comprises a GetPendingPresentationDescriptor method to retrieve the next pending presentation.

5. (Previously Presented) The computer-readable storage medium of claim 4 wherein the GetPendingPresentationDescriptor method includes a pending presentation descriptor.

6. (Previously Presented) The computer-readable storage medium of claim 1 wherein the Initialize method includes parameters, the parameters comprising:

a muxed stream descriptor;

a selected media type for the muxed stream descriptor;

an array of major types of elementary streams; and

a count of major types in the array of major types.

7. (Currently Amended) ~~The computer-readable storage medium of claim 1 wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object.~~

A computer implemented method for implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, comprising:

an Initialize method configuring the demultiplexer object;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists merely if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, and wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a pointer to a sample object;

a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor, wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object; and

a Flush method flushing currently queued input and output samples.

8. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessInput method includes a pointer to a sample object.

9. (Currently Amended) The computer-readable storage medium of claim 8 wherein the ProcessInput method ~~further~~ includes a return value having a new presentation flag.

10. (Currently Amended) The computer-readable storage medium of claim 9 ~~having further computer-executable instructions for performing the~~ method steps comprising:

if the new presentation flag has a TRUE value:

calling a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer's input queue.

11. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object.

12. (Currently Amended) The computer-readable storage medium of claim 11 wherein the ProcessOutput method ~~further~~ includes an output return value.

13. (Previously Presented) The computer-readable storage medium of claim 12 wherein the output return value includes one of an end of stream error code and a no more data error code.

14. (Currently Amended) ~~The computer-readable storage medium of claim 1 wherein the interface takes multiplexed data as an in-memory buffer of data.~~

A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method comprising:

implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, wherein the interface takes multiplexed data as an in-memory buffer of data, the interface comprising:

an Initialize method configuring the demultiplexer object;

a SetPresentationDescriptor method to dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists merely if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a return value having a new presentation flag;

a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor;

a Flush method flushing currently queued input and output samples;

a GetPresentationDescriptor method to retrieve a clone of the currently active presentation descriptor on the demultiplexer object; and

a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

if the new presentation flag has a TRUE value:

calling a GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer's input queue.

15. (Previously Presented) The computer-readable storage medium of claim 14 wherein the multiplexed data has a format comprising at least one of Digital Video, MPEG2, and ASF.

16. (Currently Amended) A The computer-readable storage medium of claim 1, the method steps further comprising:

storing an Initialize data structure for use in a demultiplexer, comprising including:

a first field containing a header;

a second field containing a muxed stream descriptor;

a third field containing a selected media type of the muxed stream descriptor;

a fourth field containing an array of major types of elementary streams;

and

a fifth field containing a count of major types in the array of major types.

17. (Currently Amended) A The computer-readable storage medium of claim 1, the steps ~~further~~ comprising:

storing a SetPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a presentation descriptor.

18. (Currently Amended) A The computer-readable storage medium of claim 1, the method steps ~~further~~ comprising:

storing a GetPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a presentation descriptor.

19. (Currently Amended) A The computer-readable storage medium of claim 1, the method steps ~~further~~ comprising:

storing a GetPendingPresentationDescriptor data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a pending presentation descriptor.

20. (Currently Amended) A The computer-readable storage medium of claim 1, the method steps ~~further~~ comprising:

storing a ProcessInput data structure for use in a demultiplexer, comprising:

a first field containing a header; and

a second field containing a pointer to a sample object.

21. (Currently Amended) A The computer-readable storage medium of claim 1, the method ~~steps~~ further comprising:

storing a ProcessOutput data structure for use in a demultiplexer, comprising:

- a first field containing a header;
- a second field containing a stream identifier; and
- a third field containing a pointer to a point to a sample object.